Amendments to the Claims

Please amend the claims such that the results are:

1. (Currently amended) An integrated circuit that processes a communication packets ~~packet~~, the integrated circuit comprising:

a core processor configured to execute a software application that directs the core processor to process the communication packets ~~packet~~;

a co-processor comprising a plurality of context buffers for storing context associated with events wherein each of said context buffers having an in-use counter indicating the number of events associated with the contents of said buffer,

the co-processor configured to automatically store context for each packet into the context buffers wherein first scheduling parameters are part of the context; and

scheduling circuitry configured to retrieve the first scheduling parameters stored in the context buffer for each ~~the~~ packet and execute a first algorithm based on the first scheduling parameters to schedule subsequent transmission of the communication packet; ~~wherein the first scheduling parameters are automatically stored by the co-processor in the context buffer~~.

2. (Previously amended) The integrated circuit of claim 1 wherein the scheduling circuitry is configured to retrieve second scheduling parameters stored in the context buffer, and concurrently with the execution of the first algorithm, execute a second algorithm based on the second scheduling parameters to schedule the subsequent transmission of the communication packet.

3. (Previously amended) The integrated circuit of claim 2 wherein the first algorithm and the second algorithm comprise guaranteed cell rate algorithms.

4. (Original) The integrated circuit of claim 1 wherein scheduling circuitry is configured to update the first scheduling parameters and write the updated scheduling parameters to the context buffer.

5. (Previously amended) The integrated circuit of claim 1 wherein the first scheduling parameters indicate a first priority level and a second priority level, wherein the first priority level has a higher priority than the second priority level, and wherein the scheduling circuitry is configured to first attempt to schedule the transmission of the communication packet with the second priority level, and if unsuccessful, then to attempt to schedule the transmission of the communication packet with the first priority level.

6. (Canceled).

7. (Currently amended) The integrated circuit of claim ~~4~~ 5 wherein the highest priority level is for scheduling constant bit rate traffic.

8. (Previously amended) The integrated circuit of claim 7 wherein a lowest priority level is for scheduling available bit rate traffic.

9. (Previously amended) The integrated circuit of claim 1 wherein a first priority level is for scheduling real-time traffic, a second priority level is for scheduling non-real-time traffic, and wherein the first priority level has a higher priority than the second priority level.

10. (Original) The integrated circuit of claim 1 wherein the scheduling circuitry is configured to operate in parallel with the core processor.

11. (Currently amended) A method of operating an integrated circuit to process ~~a~~ communication packets ~~packet~~, the method comprising:

in a core processor, executing a software application that directs the core processor to process the communication packets ~~packet~~;

in a co-processor, storing context associated with events in a plurality of context buffers, wherein each of said context buffers having an in-use counter indicating the number of events associated with the contents of said buffer,

in the co-processor, automatically storing context for each packet into the context buffers wherein first scheduling parameters are part of the context; and

in scheduling circuitry, retrieving first scheduling parameters stored ~~cached~~ in the context buffer for each ~~the~~ packet and executing a first algorithm based on the first scheduling parameters to schedule subsequent transmission of the communication packet~~, wherein the first scheduling parameters are automatically stored by the co-processor in the context buffer~~.

12. (Original) The method of claim 11 further comprising, in the scheduling circuitry, retrieving second scheduling parameters cached in the context buffer, and concurrently with the execution of the first algorithm, executing a second algorithm based on the second scheduling parameters to schedule the subsequent transmission of the communication packet.

13. (Original) The method of claim 12 wherein the first algorithm and the second algorithm comprise guaranteed cell rate algorithms.

14. (Original) The method of claim 11 further comprising, in the scheduling circuitry, updating the first scheduling parameters and writing the updated scheduling parameters to the context buffer.

15. (Previously amended) The method of claim 11 wherein the first scheduling parameters indicate a first priority level and a second priority level, wherein the first priority level has a higher priority than the second priority level, and wherein executing the first algorithm based on the first scheduling parameters to schedule the subsequent transmission of the communication packet comprises first attempting to schedule the transmission of the communication packet with the second priority level, and if unsuccessful, then attempting to schedule the transmission of the communication packet with the first priority level.

16. (Canceled).

17. (Currently amended) The method of claim ~~11~~ 15 wherein the highest priority level is for scheduling constant bit rate traffic.

18. (Previously amended) The method of claim 17 wherein a lowest one of the priority levels is for scheduling available bit rate traffic.

19. (Previously amended) The method of claim 11 wherein a first priority level is for scheduling real-time traffic, a second priority level is for scheduling non-real-time traffic, an wherein the first priority level has a higher priority than the second priority level.

20. (Original) The method of claim 11 wherein the scheduling circuitry operates in parallel with the core processor.